Minimization of the Number of Edges in an EVMDD by Variable Grouping for Fast Analysis of Multi-State Systems

Shinobu Nagayama[*]   Tsutomu Sasao[†]   Jon T. Butler[‡]

[*]*Dept. of Computer and Network Eng., Hiroshima City University, Hiroshima, JAPAN*
[†]*Dept. of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka, JAPAN*
[‡]*Dept. of Electr. and Comp. Eng., Naval Postgraduate School, Monterey, CA USA*

Abstract—This paper proposes an algorithm to minimize the number of edges in an edge-valued multi-valued decision diagram (EVMDD) for fast analysis of multi-state systems. We minimize the number of edges by grouping multi-valued variables into larger-valued variables. By grouping multi-valued variables, we can also reduce the number of nodes. However, minimization of the number of nodes by grouping variables is not always effective for fast analysis of multi-state systems. On the other hand, minimization of the number of edges is effective. Experimental results show that the proposed algorithm for minimizing the number of edges reduces the number of edges by up to 15% and the number of nodes by up to 47%. This results in a speed-up of the analysis of multi-state systems by about three times.

Keywords-Minimization algorithm of the number of edges; EVMDDs; grouping variables for optimization of decision diagrams; multi-state systems; system analysis based on decision diagrams.

I. INTRODUCTION

A multi-state system is a system model in which performance, reliability, safety, efficiency, power consumption, etc. are represented by states. It is widely used to model various fault tolerant systems including computer server systems, telecommunication systems, water, gas, electrical power distribution systems, flight control systems, and nuclear power plant monitoring systems [2], [3], [16], [20], [22]. To design dependable fault tolerant systems, intensive analysis of multi-state systems is indispensable. Since this is very time-consuming, many analysis methods have been proposed to shorten analysis time. Among them, methods based on binary decision diagrams (BDDs) [1], [2], [4], [22] and multi-valued decision diagrams (MDDs) [9], [15], [19], [20] have attracted much attention, since they hold promise as fast analysis methods.

In analysis methods based on decision diagrams (DDs), optimization of DDs is very important to reduce memory size and runtime for analysis. Most existing optimization algorithms for DDs use variable reordering approaches [5]–[7], [11], [12], [17]. However, for analysis of multi-state systems in which some components (i.e., variables) have interdependent states [10], the order of some variables can be fixed. This is because conditional probabilities $P(B|A)$

are computed to analyze such systems, and $P(B|A)$ cannot be computed unless the value of A is decided prior to B. Thus, another approach that does not change the order of variables would be more robust and effective for analysis of a wide range of systems.

In this paper, we use a variable grouping approach for optimization of DDs [13]. In many uses of DDs, minimization of the number of nodes is the objective of optimization. However, minimization of the number of nodes by grouping variables is trivial, and it is not always effective for fast analysis of multi-state systems. Thus, we propose an algorithm to minimize *the number of edges* in an edge-valued multi-valued decision diagram (EVMDD) [14], [15] by grouping multi-valued variables to larger-valued variables. By grouping variables, we can reduce not only the number of edges, but also the number of nodes effectively, resulting in faster analysis of multi-state systems.

This paper is organized as follows: Section II defines multi-state systems, EVMDDs, and variable grouping. Section III introduces the analysis method of multi-state systems using EVMDDs, and in Section IV, we propose an algorithm to minimize the number of edges in an EVMDD. Experimental results are shown in Section V.

II. PRELIMINARIES

This section defines multi-state systems, structure functions, EVMDDs to represent structure functions, and variable grouping.

A. Multi-State Systems and Structure Functions

Definition 1: A **multi-state system** is a model of a system that represents, as states, a capability, such as performance, capacity, or reliability. There are usually more than two states, and so a multiple-valued analysis is required. When components in a system are modeled as well, it is called a **multi-state system with multi-state components**. In this paper, it is simply called a multi-state system.

Definition 2: A state of a multi-state system depends only on states of components in the system. A system with n components can be considered as a multi-valued function $f(x_1, x_2, \ldots, x_n) : R_1 \times R_2 \times \ldots \times R_n \rightarrow M$, where each x_i

Report Documentation Page

1. REPORT DATE **MAY 2013**	2. REPORT TYPE	3. DATES COVERED **00-00-2013 to 00-00-2013**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Minimization of the Number of Edges in an EVMDD by Variable Grouping for Fast Analysis of Multi-State Systems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School (NPS),Department of Electrical and Computer Engineering,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Proceedings of the 43rd International Symposium on Multiple-Valued Logic, Toyama, Japan, May 21-24, 2013, pp. 284-289.

14. ABSTRACT
This paper proposes an algorithm to minimize the number of edges in an edge-valued multi-valued decision diagram (EVMDD) for fast analysis of multi-state systems. We minimize the number of edges by grouping multi-valued variables into larger-valued variables. By grouping multivalued variables, we can also reduce the number of nodes. However, minimization of the number of nodes by grouping variables is not always effective for fast analysis of multi-state systems. On the other hand, minimization of the number of edges is effective. Experimental results show that the proposed algorithm for minimizing the number of edges reduces the number of edges by up to 15% and the number of nodes by up to 47%. This results in a speed-up of the analysis of multi-state systems by about three times.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

x_1	x_2	x_3	x_4	f
0	0	0	0	0
0	0	0	1	0
0	0	0	2	0
0	0	1	0	0
0	0	1	1	1
		$\vdots$		$\vdots$
2	2	2	2	5

(a) Multi-state system.   (b) Structure function.
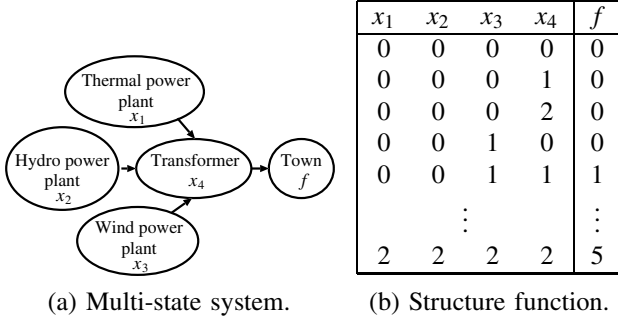
Figure 1.   Multi-state system for an electrical power distribution system and its structure function.

Figure 2.   MDD for the structure function.

Figure 3.   EVMDD for the structure function.

represents a component with r_i states, $R_i = \{0, 1, \ldots, r_i - 1\}$ is a set of the states, and $M = \{0, 1, \ldots, m - 1\}$ is a set of the m system states. This multi-valued function is called a **structure function** of the multi-state system.

Definition 3: A structure function $f(x_1, x_2, \ldots, x_n)$ is **monotone increasing** iff, for all $\alpha, \beta \in R_i$, where $\alpha \le \beta$,

$$f(x_1, x_2, \ldots, x_{i-1}, \alpha, x_{i+1}, \ldots, x_n)$$
$$\le f(x_1, x_2, \ldots, x_{i-1}, \beta, x_{i+1}, \ldots, x_n).$$

Structure functions are often monotone increasing, as illustrated by the following example.

Example 1: Fig. 1(a) shows a multi-state system for an electrical power distribution system. In this figure, the power plants x_1, x_2, x_3 and the transformer x_4 have three states which correspond to supply levels: 0 (breakdown), 1 (partially supply), and 2 (fully supply). And, the system has six states which correspond to the percentage of area of a town that is blacked out: 0 (complete blackout), 1 (90% blackout), 2 (60% blackout), 3 (30% blackout), 4 (10% blackout), and 5 (0% blackout).

In this way, by assigning a value to each state in ascending order, we obtain the 6-valued monotone increasing structure function f shown in Fig. 1(b). The table that is partially shown in Fig. 1(b) has 81 rows and is too large to be included in its entirety. However, its contents can be determined by the function's representation as an MDD or EVMDD, as discussed in the next section. (End of Example)

B. Edge-Valued Multi-Valued Decision Diagrams

Definition 4: A **multi-valued decision diagram (MDD)** is a rooted directed acyclic graph representing a multi-valued function f. The MDD is obtained by repeatedly applying the Shannon expansion to the multi-valued function [8]. It consists of non-terminal nodes representing sub-functions obtained from f by assigning values to certain variables. It also has terminal nodes representing function values. Each non-terminal node has multiple outgoing edges that correspond to the values of a multi-valued variable. The MDD is ordered; i.e., the order of variables along any path
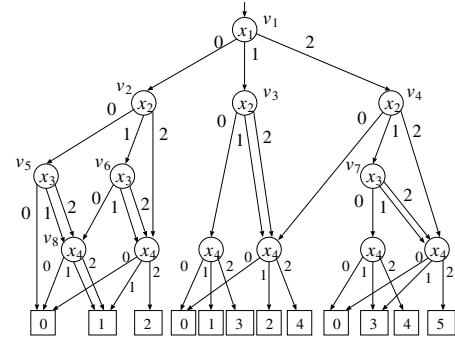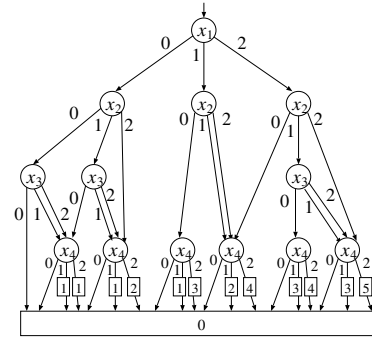
from the root node to a terminal node is the same. When an MDD represents a function for which multi-valued variables have different domains, it is a heterogeneous MDD [13]. In the following, the term 'MDD' refers to a heterogeneous MDD.

Definition 5: An **edge-valued MDD (EVMDD)** [14] is an extension of the MDD, and represents a multi-valued function. It consists of one terminal node representing 0 and non-terminal nodes with edges having integer weights; 0-edges always have zero weights. In an EVMDD, the function value is represented as a sum of weights for edges traversed from the root node to the terminal node.

Example 2: Fig. 2 and Fig. 3 show an ordinary MDD and an EVMDD for the structure function of Example 1. For readability, some terminal nodes in the MDD are not combined. (End of Example)

C. Variable Grouping

Definition 6: Let $X = (x_1, x_2, \ldots, x_n)$ be an ordered set of n multi-valued variables. Let

$$X_1 = (x_1, x_2, \ldots, x_{k_1}),$$
$$X_2 = (x_{k_1+1}, x_{k_1+2}, \ldots, x_{k_1+k_2}),$$
$$\vdots$$
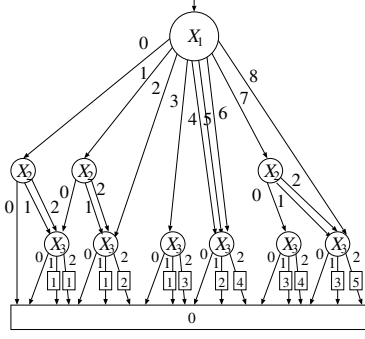
Figure 4. GEVMDD for the function $g(X_1, X_2, X_3)$.

Figure 5. Analysis of the multi-state system using EVMDD.

$$X_u = (x_{k_1+k_2+\dots+1}, x_{k_1+k_2+\dots+2}, \dots, x_n).$$

Then, $(X_1, X_2, \dots, X_u)$ is a **grouping** of X. Each ordered set $X_i = (x_{j+1}, x_{j+2}, \dots, x_{j+k_i})$ forms a **super variable** whose domain is $\{0, 1, \dots, r_{j+1} \times r_{j+2} \times \dots \times r_{j+k_i} - 1\}$, where $|X_i| = k_i$ and $k_1 + k_2 + \dots + k_u = n$. Note that the order of the original multi-valued variables is preserved in a grouping.

By considering each super variable X_i as a larger-valued variable, the original multi-valued function $f(x_1, x_2, \dots, x_n) : R_1 \times R_2 \times \dots \times R_n \to M$ can be converted into its **larger-valued input function** $g(X_1, X_2, \dots, X_u) : P_1 \times P_2 \times \dots \times P_u \to M$, where $P_i = \{0, 1, \dots, r_{j+1} \times r_{j+2} \times \dots \times r_{j+k_i} - 1\}$.

In this paper, for convenience, an EVMDD representing the function g obtained by grouping variables is called a **GEVMDD**.

Example 3: When the multi-valued variables x_1, x_2, x_3, x_4 in Example 1 are grouped into three super variables, we have

$$
\begin{aligned}
X_1 &= (x_1, x_2), \\
X_2 &= (x_3), \text{ and} \\
X_3 &= (x_4).
\end{aligned}
$$

Note that since x_1 and x_2 are 3-valued variables, the super variable X_1 consisting of x_1 and x_2 is a 9-valued variable. The GEVMDD representing the obtained function $g(X_1, X_2, X_3)$ is shown in Fig. 4. (End of Example)

III. ANALYSIS METHOD USING EVMDDS

Definition 7: The probability that a structure function f has the value s is denoted by $P_s(f = s)$, where $s \in \{0, 1, \dots, m-1\}$. The probability that a component x_i has the value c is denoted by $P_c(x_i = c)$, where $c \in \{0, 1, \dots, r_i - 1\}$.

An analysis of multi-state systems solves the following:

Problem 1: Given a structure function f of a multi-state system and the probability of each state of each component $P_c(x_i = c)$, compute the probability of each state of the multi-state system $P_s(f = s)$. For simplicity, we assume that the probabilities of all component states are independent of each other.
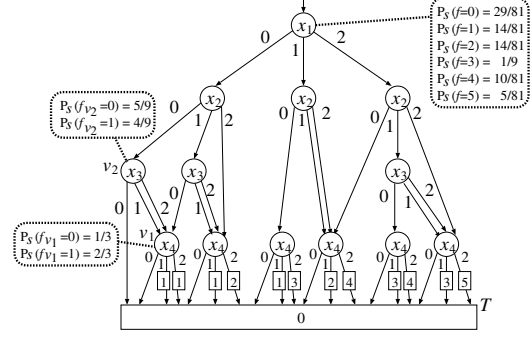
To solve this problem efficiently, a method using EVMDDs has been proposed [15]. The method represents given structure functions using EVMDDs, and computes probabilities for a structure function by merging probabilities for sub-functions represented by nodes in a bottom-up manner.

Example 4: Let us compute the probability of each state of the multi-state system using the EVMDD in Fig. 5. In this example, we assume that all states of each component occur with the same probability, $1/3$.

First, we have $P_s(f_T = 0) = 1$ at the terminal node T. Then, we compute probabilities for a sub-function f_{v_1} at node v_1. Since this node has two edges pointing to T whose values are 1, and the two edges represent $f_{v_1} = 1$, we have

$$
\begin{aligned}
P_s(f_T = 0) \times P_c(x_4 = 1) &= 1/3, \\
P_s(f_T = 0) \times P_c(x_4 = 2) &= 1/3, \text{ and thus,} \\
P_s(f_{v_1} = 1) &= P_s(f_T = 0) \times P_c(x_4 = 1) \\
&\quad + P_s(f_T = 0) \times P_c(x_4 = 2) \\
&= 2/3.
\end{aligned}
$$

Thus, $P_s(f_{v_1} = 0) = 1/3$ and $P_s(f_{v_1} = 1) = 2/3$ for v_1. At v_2, the probabilities at the terminal node and v_1 are multiplied by $1/3$, and they are merged. Thus, $P_s(f_{v_2} = 0) = 5/9$ and $P_s(f_{v_2} = 1) = 4/9$. Similarly, by performing the same computation at each node in a bottom-up manner, we have the following at the root node: $P_s(f = 0) = 29/81$, $P_s(f = 1) = 14/81$, $P_s(f = 2) = 14/81$, $P_s(f = 3) = 1/9$, $P_s(f = 4) = 10/81$, and $P_s(f = 5) = 5/81$. (End of Example)

Since in many applications, structure functions are mono-tone increasing, the functions are compactly represented by EVMDDs, and Problem 1 can be solved efficiently by an algorithm whose time complexity is $O(N_E)$, where N_E is the number of nodes in an EVMDD. However, this time complexity is a very rough estimate.

We can minimize the number of nodes in an EVMDD straightforwardly by grouping all n multi-valued variables of a given structure function into a super variable as shown
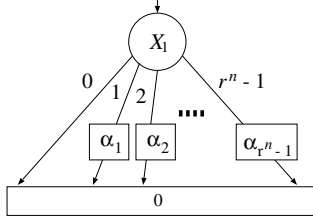
Figure 6.   GEVMDD for a function $g(X_1)$, $X_1 = (x_1, x_2, \ldots, x_n)$.

Algorithm 1: (Minimization of the number of edges)
```
1: minimize_edge (EVMDD, the number of variables n) {
2:    for(i = n; i > 0; i = i - 1) {
3:       min_edges = ∞ :
4:       for(k = 1; k ≤ limit[i]; k = k + 1) {
5:          n_edges = nodes(EVMDD, i, k) × ∏_{j=0}^{k-1} r_{i+j} ;
6:          n_edges = n_edges + lower_edges[i + k] ;
7:          if (min_edges > n_edges) {
8:             min_edges = n_edges ;
9:             register the grouping k ;
10:          }
11:       }
12:       lower_table[i] = min_edges ;
13:    }
14:    return lower_table[1] ;
15:}
```

in Fig. 6. In this case, although the number of nodes is only one, we have to access all r^n edges, and merge their probabilities, where r is the number of states for each multi-valued variable. This allows the computation of the probabilities of the function ($P_s(f = s)$). Therefore, the time complexity is, more specifically, (overhead for merging probabilities) × (the number of edges in each node) × (the number of nodes). Since the overhead for merging probabilities is small, the time complexity is $O(r^n)$ in this example.

In the optimization of DDs based on variable reordering, minimization of the number of nodes is effective for fast analysis since the number of edges in each node is constant. However, in the optimization of EVMDDs based on variable grouping, minimization of the number of edges in an EVMDD is more effective.

IV. MINIMIZATION OF THE NUMBER OF EDGES

Example 5: The EVMDD shown in Fig. 3 has 39 edges. On the other hand, the GEVMDD shown in Fig. 4 has 36 edges, and it is the GEVMDD with the minimum number of edges. If the variables x_1, x_2, x_3, and x_4 are grouped into a single super variable as in Fig. 6, then a GEVMDD obtained by this grouping has $3^4 = 81$ edges.   (End of Example)

In this way, different groupings of variables produce GEVMDDs with a different number of edges. Thus, there is an optimum grouping of variables that produces a GEVMDD with the minimum number of edges. This section formulates a minimization problem of the number of edges in an EVMDD, and then presents a minimization algorithm.

Problem 2: Given an EVMDD representing a structure function $f(x_1, x_2, \ldots, x_n)$, find a grouping of variables $(x_1, x_2, \ldots, x_n)$ that produces a GEVMDD with the minimum number of edges.

Algorithm 1 shows a pseudo-code to solve Problem 2. This algorithm is based on dynamic programing, and searches for the minimum number of edges for each sub-EVMDD sequentially from the bottom. In the following, for simplicity, we assume that the variable order for a given EVMDD is $x_1, x_2, \ldots, x_n$ from the top to the bottom.

Algorithm 1 is efficient because **limit[i]** prevents unnecessary iterations of the second for loop. This is shown by the following theorem.

Theorem 1: Let **nodes(EVMDD, *i*, *k*)** be the number of nodes in a GEVMDD with respect to a super variable that consists of k variables from x_i to x_{i+k-1}, and let **edges(EVMDD, *i*)** be the number of edges associated with nodes in the given EVMDD representing variables from x_i to x_n. If, for some value of k, the following relation holds:

$$\textbf{nodes(EVMDD, } i, k) \times \prod_{j=0}^{k-1} r_{i+j} > \textbf{edges(EVMDD, } i),$$

then for any $k' \geq k$, the same relation holds:

$$\textbf{nodes(EVMDD, } i, k') \times \prod_{j=0}^{k'-1} r_{i+j} > \textbf{edges(EVMDD, } i).$$

(Proof) See Appendix.

This theorem states that, once the number of edges in a GEVMDD becomes larger than that in an EVMDD, the number of edges in a GEVMDD never becomes smaller, even if the number of variables in a super variable increases. Thus, we can prune such redundant branching.

In the 5th line, **nodes(EVMDD, i, k)** denotes the number of root nodes for sub-EVMDDs from x_i to x_{i+k-1}. When k variables $x_i, x_{i+1}, \ldots, x_{i+k-1}$ are grouped into a super variable, each root node for the sub-EVMDDs corresponds to each node in a GEVMDD with respect to the super variable, which has $\prod_{j=0}^{k-1} r_{i+j}$ edges. That is, the 5th line computes the number of edges in the GEVMDD with respect to the super variable from x_i to x_{i+k-1}.

In the 6th line, the table **lower_edges[i + k]** stores the minimum number of edges computed for the lower-EVMDD from x_{i+k} to x_n. By summing this number and the number of edges computed in the 5th line, we have the number of edges in sub-EVMDDs from x_i to x_n.

The time complexity of Algorithm 1 is $O(n^2)$. However, the coefficient of n^2 is very small due to Theorem 1.

Table I
EVMDDs AND GEVMDDs FOR *m*-STATE SYSTEMS WITH *n* 3-STATE COMPONENTS.

n	*m*	Number of nodes			Number of edges			Computation time (*μ*sec.)		
		EVMDD	GEVMDD	Ratio	EVMDD	GEVMDD	Ratio	EVMDD	GEVMDD	Ratio
5	3	10	8	80%	27	27	100%	1.20	1.08	90%
5	10	18	15	83%	51	48	94%	2.52	2.37	94%
10	3	15	13	87%	42	42	100%	1.85	1.83	99%
10	10	57	47	82%	168	162	96%	7.84	7.20	92%
10	100	265	171	65%	792	750	95%	49.27	35.54	72%
10	1,000	907	547	60%	2,718	2,364	87%	317.79	181.73	57%
15	3	30	28	93%	87	87	100%	3.72	3.84	103%
15	10	105	102	97%	312	309	99%	14.07	13.35	95%
15	100	708	613	87%	2,121	2,076	98%	110.72	99.30	90%
15	1,000	3,362	2,472	74%	10,083	9,597	95%	744.59	537.08	72%
15	10,000	11,474	8,219	72%	34,419	31,212	91%	4,701.00	2,504.00	53%
15	100,000	62,759	33,575	53%	188,274	159,768	85%	60,901.00	17,996.00	30%

n: Number of 3-state components.   *m*: Number of states for systems.   Ratio: GEVMDD / EVMDD $\times$ 100 (%)
The computation time is an average time obtained by running the same computation $1,000,000$ times, and dividing its total
time by $1,000,000$.

Since the proposed algorithm does not change the order of the original variables, it can be also applied to the analysis of multi-state systems in which some components have interdependent states [10].

V. EXPERIMENTAL RESULTS

To show the effectiveness of the proposed optimization algorithm for fast system analysis, we used the same analysis algorithm and the same structure functions as [15]. The algorithms are implemented on our private EVMDD package, and run on the following computer environment: CPU: Intel Core2 Quad Q6600 2.4GHz, memory: 4GB, OS: CentOS 5.7, and C-compiler: gcc -O2 (version 4.1.2). Table I shows the experimental results for randomly generated *m*-state systems with *n* 3-state components.

From this table, we can see that GEVMDDs have fewer nodes than EVMDDs for all functions. Especially, as the number of states *m* becomes larger, the difference in the number of nodes between GEVMDDs and EVMDDs becomes larger. With respect to the number of edges, Table I shows a similar tendency, although the relative reduction is not so large.

Surprisingly, the computation time of the analysis of multi-state systems is reduced more than the number of edges and nodes are reduced, when *m* is large. This is because a reduction in the number of nodes and edges reduces the overhead of merging probabilities. In the analysis method using EVMDDs, probabilities of function values at each node are merged at its parent node, as shown in Fig. 5. Thus, the overhead of merging probabilities increases as the number of function values at child nodes increases. Our optimization algorithm usually groups nodes near the root node into one node, as shown in Fig. 4. Since nodes near the root node tend to have many function values, this grouping yields a significant reduction in overhead. This results in faster system analysis.

From these results, we can say that the proposed optimization algorithm is very effective for fast system analysis, since minimization of the number of edges by variable grouping reduces the number of nodes, as well as overhead for merging probabilities. Especially, when the number of states *m* is large, we can represent structure functions more compactly, and analyze multi-state systems more quickly.

VI. CONCLUSION AND COMMENTS

This paper proposes a minimization algorithm of the number of edges in an EVMDD for fast analysis of multi-state systems. The proposed algorithm minimizes the number of edges by grouping multi-valued variables into larger-valued variables. By grouping multi-valued variables, we can also reduce the number of nodes and overhead for merging probabilities. Experimental results show that the proposed algorithm reduces the number of edges by up to 15% and reduces the number of nodes by up to 47%, resulting in much faster analysis of multi-state systems.

As a future work, we will study an EVMDD-based analysis method of systems in which components have interdependent states.

REFERENCES

[1] J. D. Andrews and S. J. Dunnett, "Event-tree analysis using binary decision diagrams," *IEEE Transactions on Reliability*, Vol. 49, No. 2, pp. 230–238, June 2000.

[2] Y.-R. Chang, S. V. Amari, and S.-Y. Kuo, "Reliability evaluation of multi-state systems subject to imperfect coverage using OBDD," *Proc. of the 2002 Pacific Rim International Symposium on Dependable Computing (PRDC'02)*, pp. 193–200, 2002.

[3] S. A. Doyle, J. B. Dugan, and F. A. Patterson-Hine, "A combinatorial approach to modeling imperfect coverage," *IEEE Transactions on Reliability*, Vol. 44, No. 1, pp. 87–94, Mar. 1995.

[4] S. A. Doyle and J. B. Dugan, "Dependability assessment using binary decision diagrams (BDDs)," *25th International Symposium on Fault-Tolerant Computing (FTCS)*, pp. 249–258, June 1995.

[5] R. Drechsler, W. Günther, and F. Somenzi, "Using lower bounds during dynamic BDD minimization," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, Vol. 20 No. 1, pp. 51–57, Jan. 2001.

[6] M. Fujita, Y. Matsunaga, and T. Kakuda, "On variable ordering of binary decision diagrams for the application of multi-level logic synthesis," *EDAC*, pp. 50–54, Mar. 1991.

[7] N. Ishiura, H. Sawada, and S. Yajima, "Minimization of binary decision diagrams based on exchanges of variables," *International Conference on Computer-Aided Design (IC-CAD'91)*, pp. 472–475, Nov. 1991.

[8] T. Kam, T. Villa, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "Multi-valued decision diagrams: Theory and applications," *Multiple-Valued Logic: An International Journal*, Vol. 4, No. 1-2, pp. 9–62, 1998.

[9] T. W. Manikas, M. A. Thornton, and D. Y. Feinstein, "Using multiple-valued logic decision diagrams to model system threat probabilities," *41th International Symposium on Multiple-Valued Logic*, pp. 263–267, May 2011.

[10] T. W. Manikas, D. Y. Feinstein, and M. A. Thornton, "Modeling medical system threats with conditional probabilities using multiple-valued logic decision diagrams," *42nd International Symposium on Multiple-Valued Logic*, pp. 244–249, May 2012.

[11] D. M. Miller and R. Drechsler, "Augmented sifting of multiple-valued decision diagrams," *33rd International Symposium on Multiple-Valued Logic*, pp. 375–382, Tokyo, Japan, May 2003.

[12] S. Nagayama A. Mishchenko, T. Sasao, and J. T. Butler, "Exact and heuristic minimization of the average path length in decision diagrams," *Journal of Multiple-Valued Logic and Soft Computing*, Vol. 11, No. 5-6, pp. 437–465, Aug. 2005.

[13] S. Nagayama and T. Sasao, "On the optimization of heterogeneous MDDs," *IEEE Trans. on CAD*, Vol. 24, No. 11, pp. 1645–1659, Nov. 2005.

[14] S. Nagayama, T. Sasao, and J. T. Butler, "A systematic design method for two-variable numeric function generators using multiple-valued decision diagrams," *IEICE Trans. on Information and Systems*, Vol. E93-D, No. 8, pp. 2059–2067, Aug. 2010.

[15] S. Nagayama, T. Sasao, and J. T. Butler, "Analysis of multi-state systems with multi-state components using EVMDDs," *42nd International Symposium on Multiple-Valued Logic*, pp.122-127, May, 2012.

[16] J. E. Ramirez-Marquez and D. W. Coit, "Composite importance measures for multi-state systems with multi-state components," *IEEE Transactions on Reliability*, Vol. 54, No. 3, pp. 517–529, Sept. 2005.

[17] R. Rudell, "Dynamic variable ordering for ordered binary decision diagrams," *International Conference on Computer-Aided Design (ICCAD'93)*, pp. 42–47, Nov. 1993.

[18] T. Sasao and M. Fujita (eds.), *Representations of Discrete Functions*, Kluwer Academic Publishers 1996.

[19] L. Xing and J. B. Dugan, "Dependability analysis using multiple-valued decision diagrams," *Proc. of 6th International Conference on Probabilistic Safety Assessment and Management*, June 2002.

[20] L. Xing and Y. Dai, "A new decision-diagram-based method for efficient analysis on multistate systems," *IEEE Transactions on Dependable and Secure Computing*, Vol. 6, No. 3, pp. 161–174, 2009.

[21] S. N. Yanushkevich, D. M. Miller, V. P. Shmerko, and R. S. Stankovic, *Decision Diagram Techniques for Micro- and Nanoelectronic Design*, CRC Press, Taylor & Francis Group, 2006.

[22] X. Zang, D. Wang, H. Sun, and K. S. Trivedi, "A BDD-based algorithm for analysis of multistate systems with multistate components," *IEEE Transactions on Computers*, Vol. 52, No. 12, pp. 1608–1618, Dec. 2003.

APPENDIX

Proof for Theorem 1: Suppose that for a value of k, the following relation holds:

$$\mathbf{nodes(EVMDD,} \ i, \ k) \times \prod_{j=0}^{k-1} r_{i+j} > \mathbf{edges(EVMDD,} \ i) \ \text{(A.1)}$$

Then, we will prove that, for $k+1$, (A.1) also holds.

By multiplying both sides of (A.1) by r_{i+k}, we have

$$\mathbf{nodes(EVMDD,} \ i, \ k) \times \prod_{j=0}^{k-1} r_{i+j} \times r_{i+k}$$
$$> \mathbf{edges(EVMDD,} \ i) \times r_{i+k}, \quad \text{(A.2)}$$

where r_{i+k} is the number of values of x_{i+k}.

From the definition of a super variable, the number of edges in a GEVMDD with respect to a super variable that consists of $k+1$ variables from x_i to x_{i+k} is

$$\mathbf{nodes(EVMDD,} \ i, \ k+1) \times \prod_{j=0}^{k} r_{i+j}.$$

Since $\mathbf{nodes(EVMDD,} \ i, \ k)$ is monotone increasing with respect to k, we have

$$\mathbf{nodes(EVMDD,} \ i, \ k+1) \geq \mathbf{nodes(EVMDD,} \ i, \ k)$$

and thus,

$$\mathbf{nodes(EVMDD,} \ i, \ k+1) \times \prod_{j=0}^{k} r_{i+j}$$
$$\geq \mathbf{nodes(EVMDD,} \ i, \ k) \times \prod_{j=0}^{k} r_{i+j}. \quad \text{(A.3)}$$

From (A.1), (A.2), and (A.3), the relation (A.1) holds for $k+1$. Therefore, for any $k' \geq k$, the theorem holds. ∎